



# Chapter 3

## วิดเจ็ตและการแสดงผล

# Outline

- Widget
- Scaffold
- Container
- Text
- AppBar
- Column
- Row
- Icon
- Image
- Center
- Padding
- ListView

# โครงสร้างโปรเจ็ค

- **lib** เก็บไฟล์ที่แยกหน้าแอปต่างๆมีนามสกุล .dart เช่น main.dart ถูกประกาศให้เป็นจุดเริ่มต้นทำงานของแอปเราโดยรันผ่านเมธอด main หรือทำเป็นรูปแบบ mvc ก็ได้
- **pubspec.yaml** สำหรับการตั้งค่าโปรเจคหรือใช้งานไลบรารีต่างๆ
- **android , ios** จัดเก็บโปรเจคของ application แต่ละระบบเอาไว้

# กำหนดชื่อแอป Android ใหม่

1. android -> app -> src -> main->AndroidManifest.xml
2. กำหนดชื่อแอปใหม่ใน android:label
3. Build
4. เสร็จสมบูรณ์



# ความละเอียดหน้าจอ

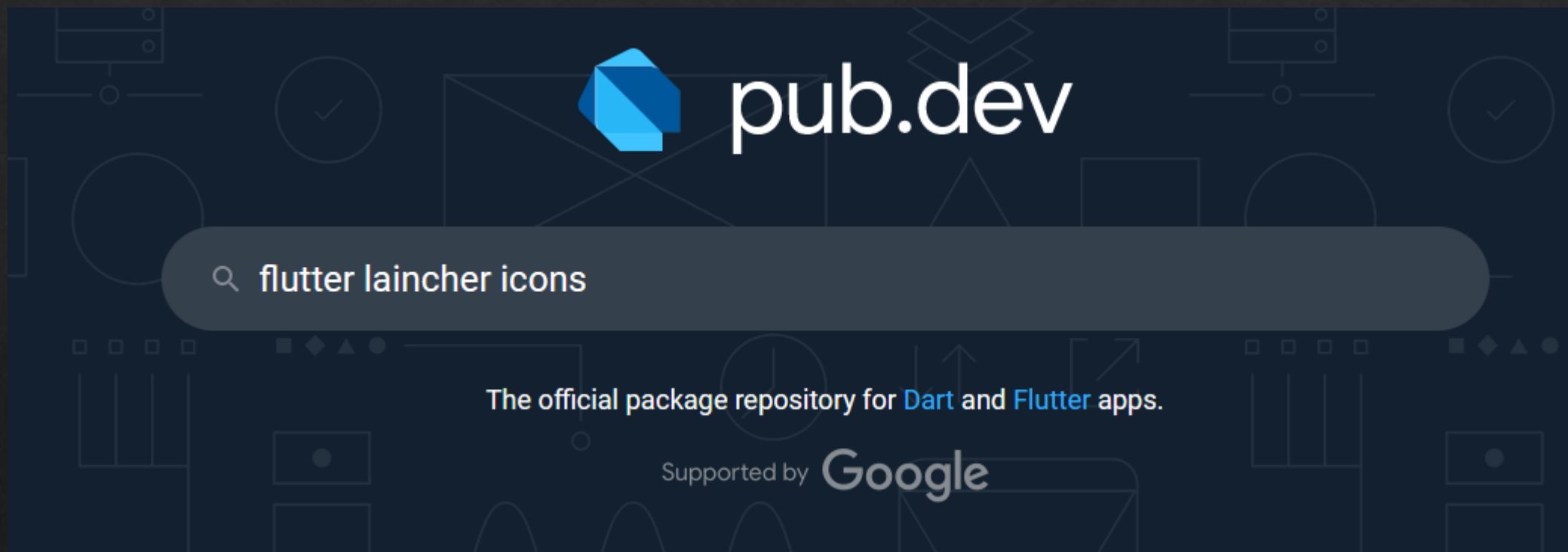
1. hdpi [high-density] - หน้าจอความละเอียดสูง
2. mdpi [medium-density] หน้าจอความละเอียดปานกลาง
3. xhdpi
4. xxhdpi
5. xxxhdpi

# กำหนดไอคอนให้แอป

1. pub.dev
2. ติดตั้ง package flutter\_launcher\_icons
3. android -> app->src->main->res [resource] -> รูปแบบความละเอียดหน้าจอ
4. โฟลเดอร์ที่แยกเก็บภาพไอคอนสำหรับใช้แสดงผลในหน้าจอที่มีความละเอียดแตกต่างกัน

# กำหนดไอคอนให้แอป

<https://pub.dev/>



```
37  ∨ dev_dependencies:  
38  ∨   flutter_test:  
39     |   sdk: flutter  
40     |   flutter_launcher_icons: ^0.9.2  
41  
42  ∨ flutter_icons:  
43     |   android: true  
44     |   ios: true  
45     |   image_path: "asset/logo.png"
```



## `flutter pub run flutter_launcher_icons:main`

- Creating default icons Android
- Overwriting the default Android launcher icon with a new icon

WARNING: Icons with alpha channel are not allowed in the Apple App Store.  
Set "remove\_alpha\_ios: true" to remove it.

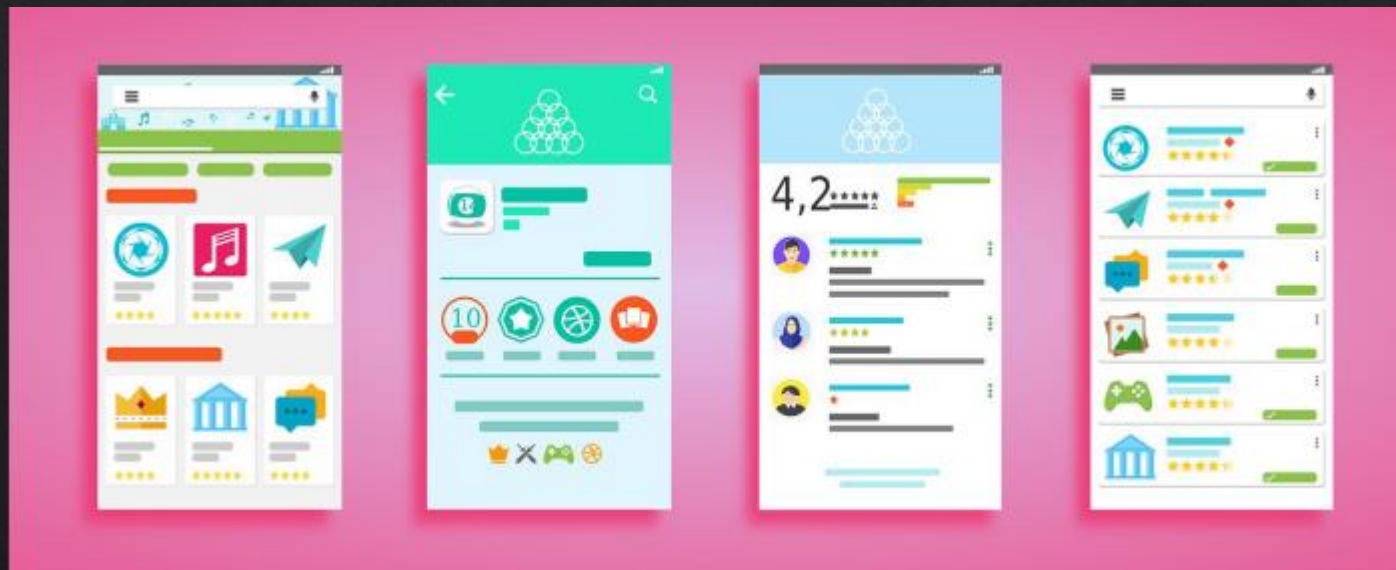
- Overwriting default iOS launcher icon with new icon

✓ Successfully generated launcher icons

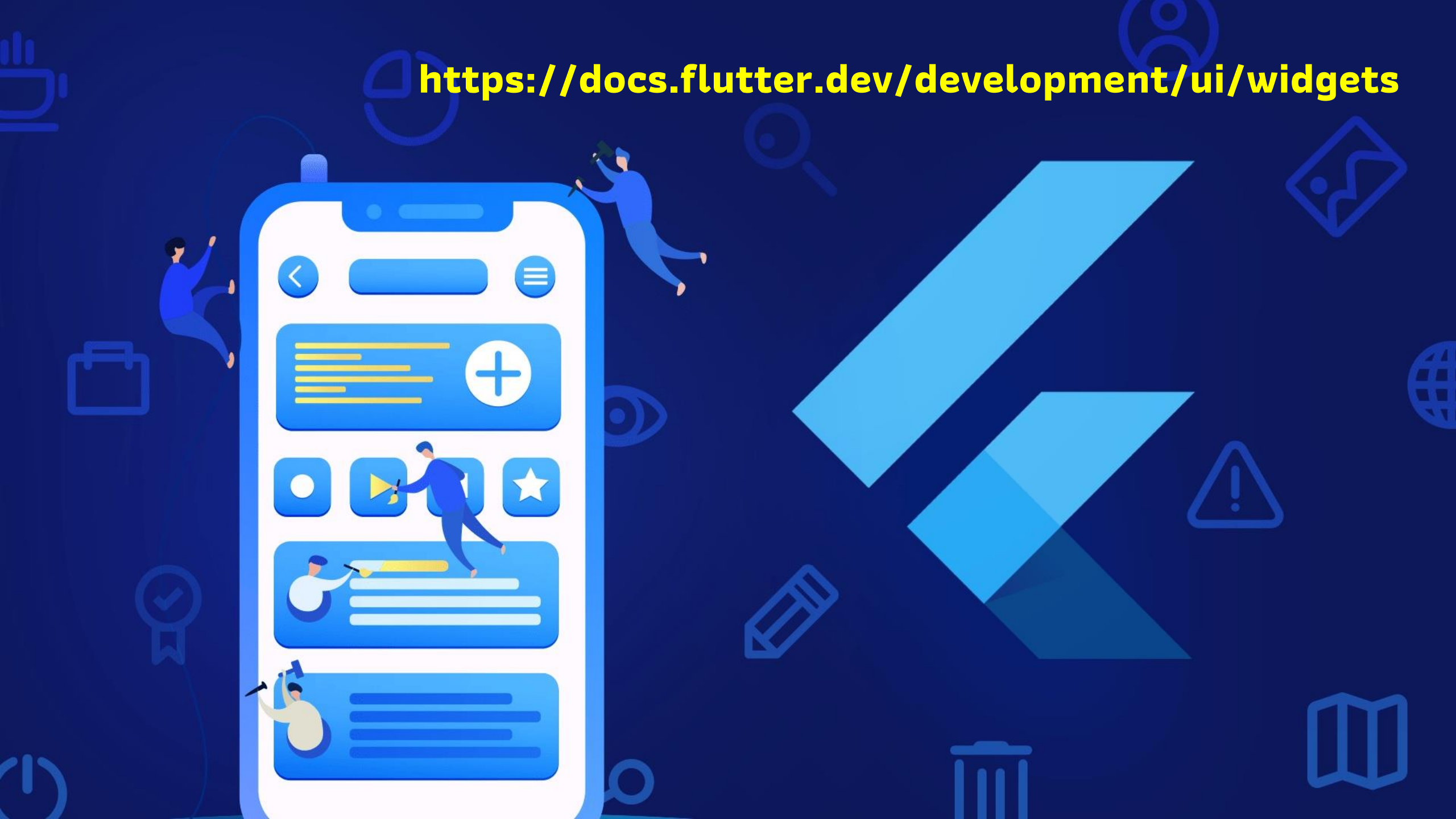
PS C:\FlutterApp\myapp>

# Widget คืออะไร

แนวคิดในการแบ่ง User Interface ออกเป็นชิ้นส่วนต่างๆ แล้วทำหน้าที่แสดงผลบนหน้าจอ เราจะเรียกชิ้นส่วนนี้ว่า **Widget** เช่น ปุ่ม ข้อความ เป็นต้น Widget แต่ละตัวจะมีส่วนที่เรียกว่า **Properties** สำหรับกำหนดคุณสมบัติให้กับ Widget นั้นๆ ซึ่ง Widget แต่ละตัวก็มี Properties ที่หลากหลายให้เราใช้งานแตกต่างกันออกไป เช่น กำหนดสี เส้นขอบ ส่วนโค้ง เป็นต้น



<https://docs.flutter.dev/development/ui/widgets>



# Widget พื้นฐานที่อยู่ในภาษา Dart

- Text Widget สำหรับข้อความ
- RaisedButton สำหรับจัดการปุ่ม
- Row , Column สำหรับการสร้าง Layout [เค้าโครงหน้าแอป] แบบแนวตั้ง และแนวนอน
- Stack สร้าง Layout แบบซ้อนทับหรือเรียงลำดับ
- Container กล่องที่รวบรวม Widget ต่างๆ

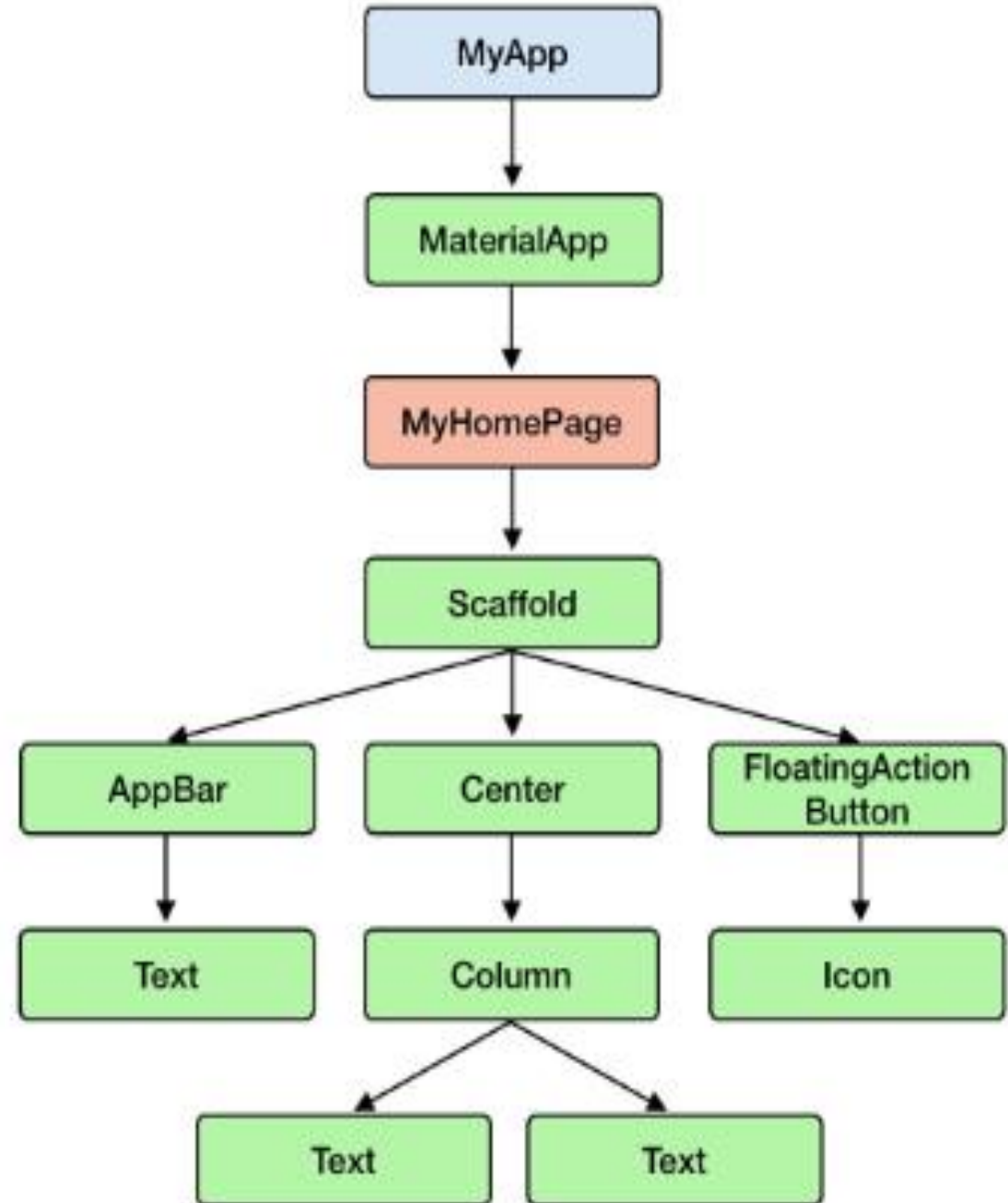
ใน Flutter จะมี Widget ที่มีชื่อว่า Material Design

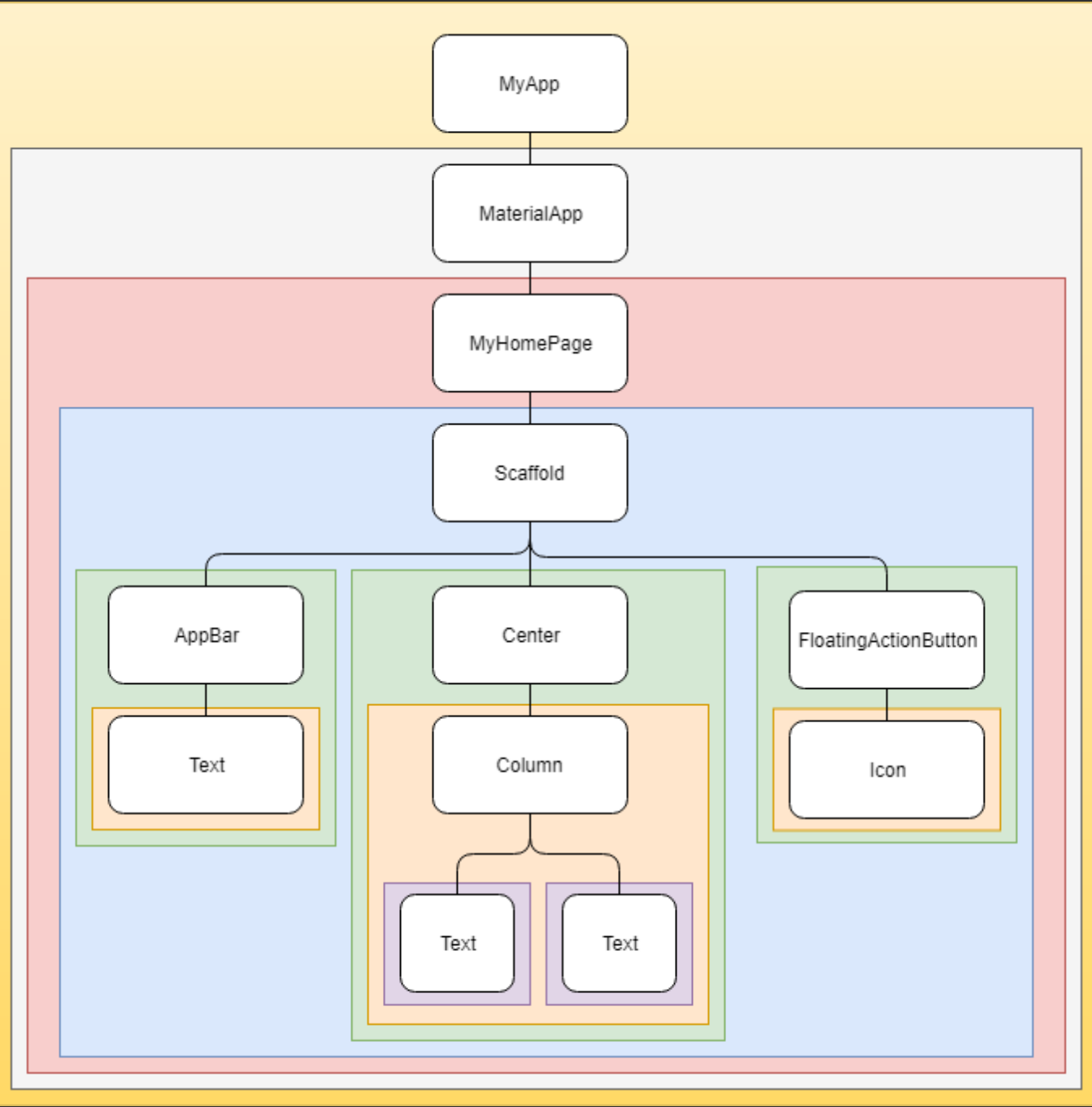
**[มาตรฐานหรือโครงสร้างในการออกแบบบนสมาร์ตโฟนและเว็บ]**

มากำหนดเป็น Layout ง่ายๆ ให้เรากำหนดชื่อ Title บน AppBar [เมนูด้านบน แอป]

# Widget Tree

คือการนำ Widget มาซ้อนทับกัน  
แบบแผนผังต้นไม้ Widget ที่บรรจุ  
Widget อื่นให้อยู่ภายในเรียกว่า  
**Parent Widget** ส่วน Widget ที่ด้าน  
ในจะเรียกว่า **Children Widget**





## **child**

จะอยู่ใน Widget ที่บรรจุ Widget ได้ตัวเดียว เช่น Center , Container

## **children**

จะอยู่ใน Widget ที่บรรจุ Widget ได้มากกว่าหนึ่งตัว  
เช่น Row, Column, ListView, Stack

`child` takes a single widget

```
child: Text('foo')
```

`children` takes a list of widgets

```
children: <Widget>[Text('foo'), Text('bar')]
```



# เริ่มต้นใช้งาน Widget

```
// ต้องการหน้าจอสวยๆ แทน console  
import 'package:flutter/material.dart';
```

```
void main(){  
  runApp [app]; //app คือการระบุ widget ต้องการทำงาน  
}
```

# เริ่มต้นใช้งาน MaterialApp Widget

MaterialApp จะใช้การควบคุมการแสดงผลในหน้าแอปของเรา

```
void main(){  
  var app = MaterialApp(title:"Hello",home:Text['My App']);  
  // สร้าง widget ขึ้นมา  
  runApp(app);  
}
```

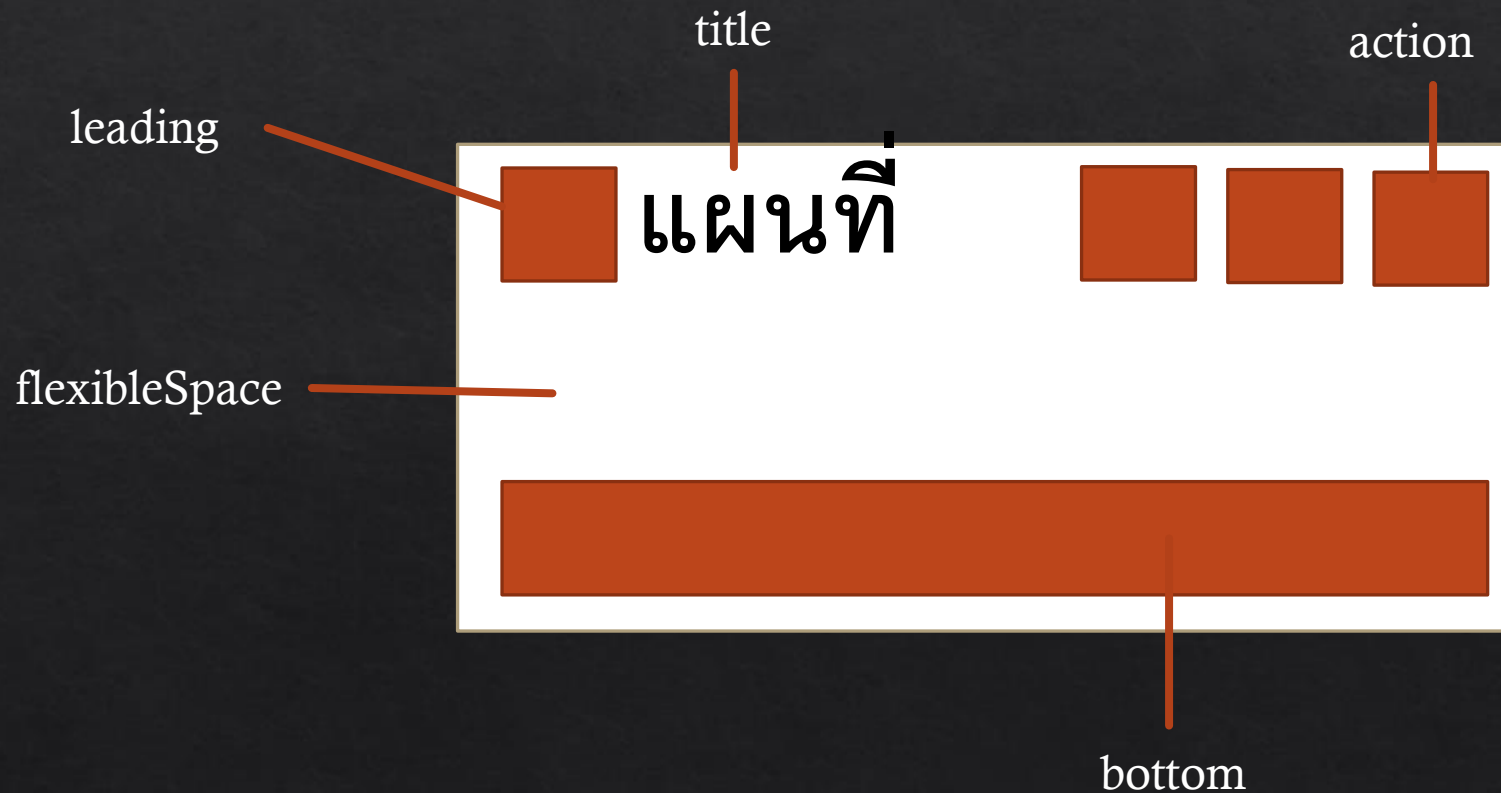
# Scaffold

Scaffold คือ Widget หน้าต่าง**สำเร็จรูป**สำหรับจัดการ Layout หรือ โครงสร้างของหน้าแอปมีการคำนวณระยะห่างของแอปกับหน้าจอ Emulator ให้อัตโนมัติ

- appBar
- body

# AppBar

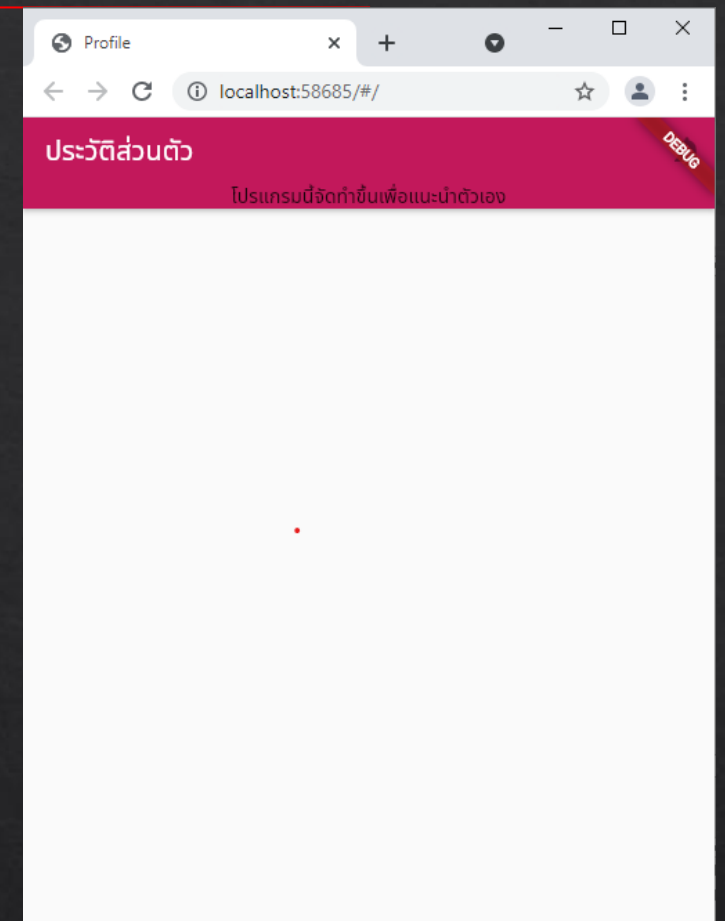
AppBar คือแถบ**เครื่องมือที่แสดงด้านบน**ของหน้าจอ ใช้เพื่อแสดงหัวข้อสำคัญ หรือเมนูการใช้งานต่างๆ



# AppBar

- title
- backgroundColor
- action
- automaticallyImplyLeading
- bottom

```
home: Scaffold(  
  appBar: AppBar(  
    title: Text('ประวัติส่วนตัว'),  
    actions: <Widget>[  
      IconButton(  
        onPressed: null,  
        icon: Icon(Icons.add_alert),  
      )  
    ],  
    automaticallyImplyLeading: false,  
    backgroundColor: Colors.pink.shade700,  
    bottom: PreferredSize(  
      preferredSize: const Size.fromHeight(10.0),  
      child: Text(  
        'โปรแกรมนี้จัดทำขึ้นเพื่อแนะนำตัวเอง',  
        textAlign: TextAlign.center,  
      ),  
    ),  
  ),  
  bottomNavigationBar: BottomAppBar(),  
),
```



# AppBar

AppBar คือแถบ**เครื่องมือที่แสดงด้านบน**ของหน้าจอ ใช้เพื่อแสดงหัวข้อสำคัญ หรือเมนูการใช้งานต่างๆ

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(),
      ));
  }
}
```

# Body

Body คือส่วนที่กำหนดเนื้อหาของการแสดงผลภายใน Scaffold สามารถเขียนเป็น attribute ต่อจาก appBar ได้โดยคั่นด้วยเครื่องหมายจุลภาค

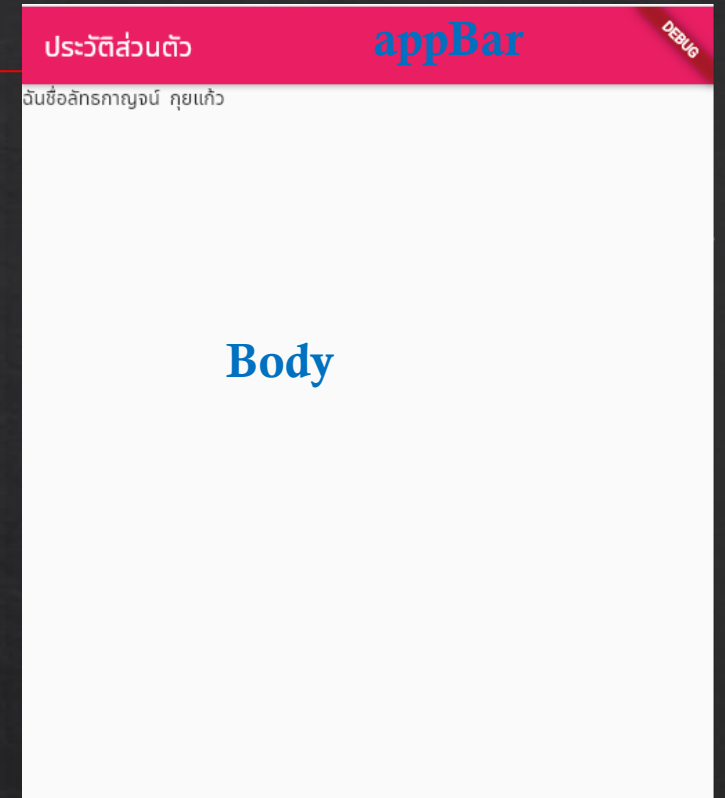
```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(),
        body: Text('สวัสดี Flutter'),
      ),
    );
  }
}
```



```
void main() {  
  var app = MaterialApp(  
    title: "Profile",  
    home: Scaffold(  
      appBar: AppBar(  
        title: Text('ประวัติส่วนตัว'),  
      ),  
      body: Text("ฉันชื่อลัทธภาญจน์ กุญแจแก้ว"),  
    ),  
    theme: ThemeData(primarySwatch: Colors.pink), //เปลี่ยนสี app  
  );  
  runApp(app);  
}
```





# TextAlign

การจัดแนวการแสดงผลของข้อความ

**textAlign.center**

- center
- justify
- left
- right

# Overflow

การกำหนดคุณสมบัติการมองเห็นในการแสดงผลข้อความที่มีจำนวนตัวอักษรมากๆ จนล้นขอบเขตของหน้า



ทดสอบข้อความ กำหนดให้ยาวกว่าความกว้าง  
หน้าจอ

● clip



ทดสอบข้อความ กำหนดให้ยาวกว่าความกว้าง...

● ellipsis

ทดสอบข้อความ กำหนดให้ยาวกว่าความกว้าง...

```
21 | child: Text(  
22 |   'ทดสอบข้อความ กำหนดให้ยาวกว่าความกว้างหน้าจอ',  
23 |   overflow: TextOverflow.ellipsis,  
24 |   style: TextStyle(  
25 |     color: Colors.black,  
26 |     backgroundColor: Colors.white,  
27 |     fontWeight: FontWeight.bold,  
28 |     fontSize: 20.0  
29 |   ) // TextStyle  
30 | ) // Text
```

# Style

การกำหนดรูปแบบสไตลข้อความ เช่น ขนาดตัวอักษร ขนาดความหนา ตัวอักษร สี ฟอนต์ เป็นต้น

```
21 child: const Text.rich(  
22   TextSpan(  
23     text: 'Hello', // ใช้ค่า default text style  
24     children: <TextSpan>[  
25       TextSpan(  
26         text: 'beautiful',  
27         style: TextStyle(  
28           color: Colors.pink,  
29           fontStyle: FontStyle.italic  
30         ) // TextStyle  
31       ), // TextSpan  
32       TextSpan(  
33         text: 'world',  
34         style: TextStyle(  
35           color: Colors.yellow,  
36           fontWeight: FontWeight.bold  
37         ) // TextStyle  
38       ), // TextSpan  
39     ], // <TextSpan>[]  
40   ), // TextSpan  
41 ) // Text.rich
```

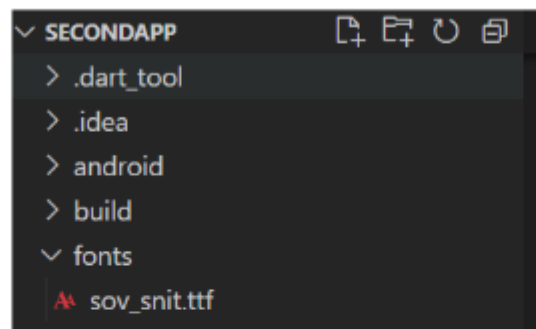
Hello *beautiful* world

# กำหนดรูปแบบ font ให้กับ text

download font



สร้าง folder



File ชื่อ pubspec.yaml

```
fonts:  
- family: sovsnit  
  fonts:  
    - asset: fonts/sov_snit.ttf
```

ใส่ fontFamily ที่สร้างใน main.dart

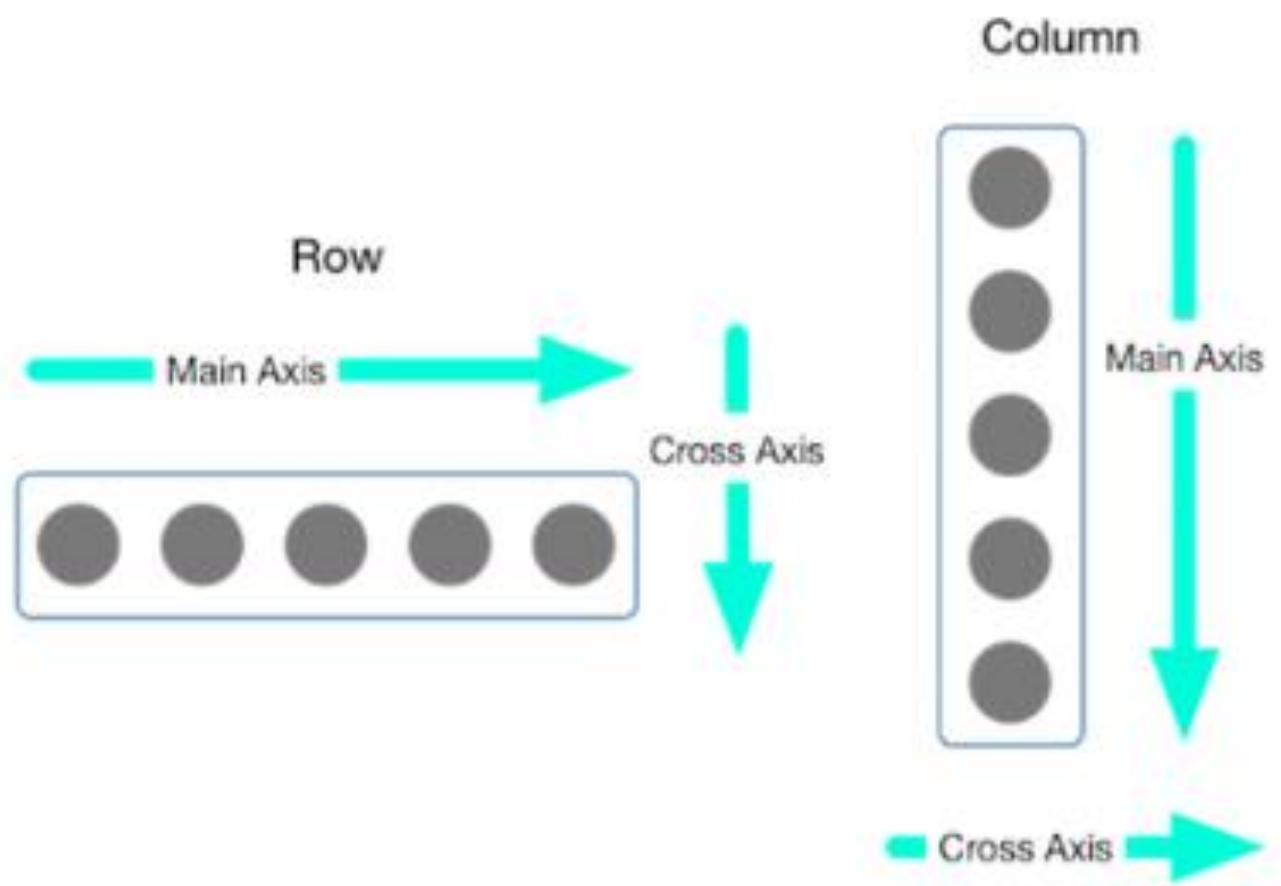
```
body: Center(child: Text("สวัสดี",  
style: TextStyle(fontSize: 100, fontWeight: FontWeight.bold,  
color: Colors.blue, fontFamily: 'sovsnit'),),), // TextStyle // Text // Center
```

# Column

เป็นวิดเจ็ต ที่ใช้ในการจัดตำแหน่งการแสดงผลแบบคอลัมน์ เพื่อแยกวัตถุ เป็นสัดส่วนไม่ชิดติดกัน

```
body: Column(  
  children: <Widget>[  
    Text('สวัสดี'),  
    Text('ฉันชื่อลัทธิกาญจน์ กุญแจ'),  
    Text('ชื่อเล่นแอน'),  
  ],  
)  
,
```





# Column

- **crossAxisAlignment**

กำหนดการจัดแนวของวัตถุทางขวาหรือแนวนอน ซึ่งสามารถกำหนดค่าการใช้งานได้เช่น

- crossAxisAlignment : CrossAxisAlignment.start
- crossAxisAlignment : CrossAxisAlignment.center
- crossAxisAlignment : CrossAxisAlignment.end

- **mainAxisAlignment**

กำหนดคุณสมบัติการจัดวางวัตถุตามแนวตั้ง ซึ่งสามารถกำหนดค่าการใช้งานได้เช่น

- mainAxisAlignment : MainAxisAlignment.start
- mainAxisAlignment : MainAxisAlignment.center
- mainAxisAlignment : MainAxisAlignment.end

# Row

เป็นวิดเจ็ต ที่ช่วยในการจัดตำแหน่งการแสดงผลของลูกๆ (children) ซึ่งช่วยทำให้วิดเจ็ตต่างๆสามารถจัดเรียงในแนวนอนได้ มักใช้ควบคู่กับการจัดแนวแบบคอลัมน์

```
body: Column(  
  children: <Widget>[  
    Row(  
      children: <Widget>[  
        Text('สวัสดี'),  
        Text('ฉันชื่อลัทธภาณูญณ์ กุญแก้ว'),  
        Text('ชื่อเล่นแอน'),  
      ],  
    ),  
  ],  
);
```

# Icon

เป็นรูปภาพกราฟฟิค ที่วาดขึ้นเพื่อใช้เป็นสัญลักษณ์เสริมกับข้อความ ช่วยให้เข้าใจสิ่งที่ต้องการศึกษาได้รวดเร็วขึ้น

- icons
- color
- size
- textDirection

**<https://material.io/resources/icons>**

```
children: const <Widget>[  
  Icon(  
    Icons.android,  
    color: Colors.green,  
    size: 24.0,  
  ),  
],
```

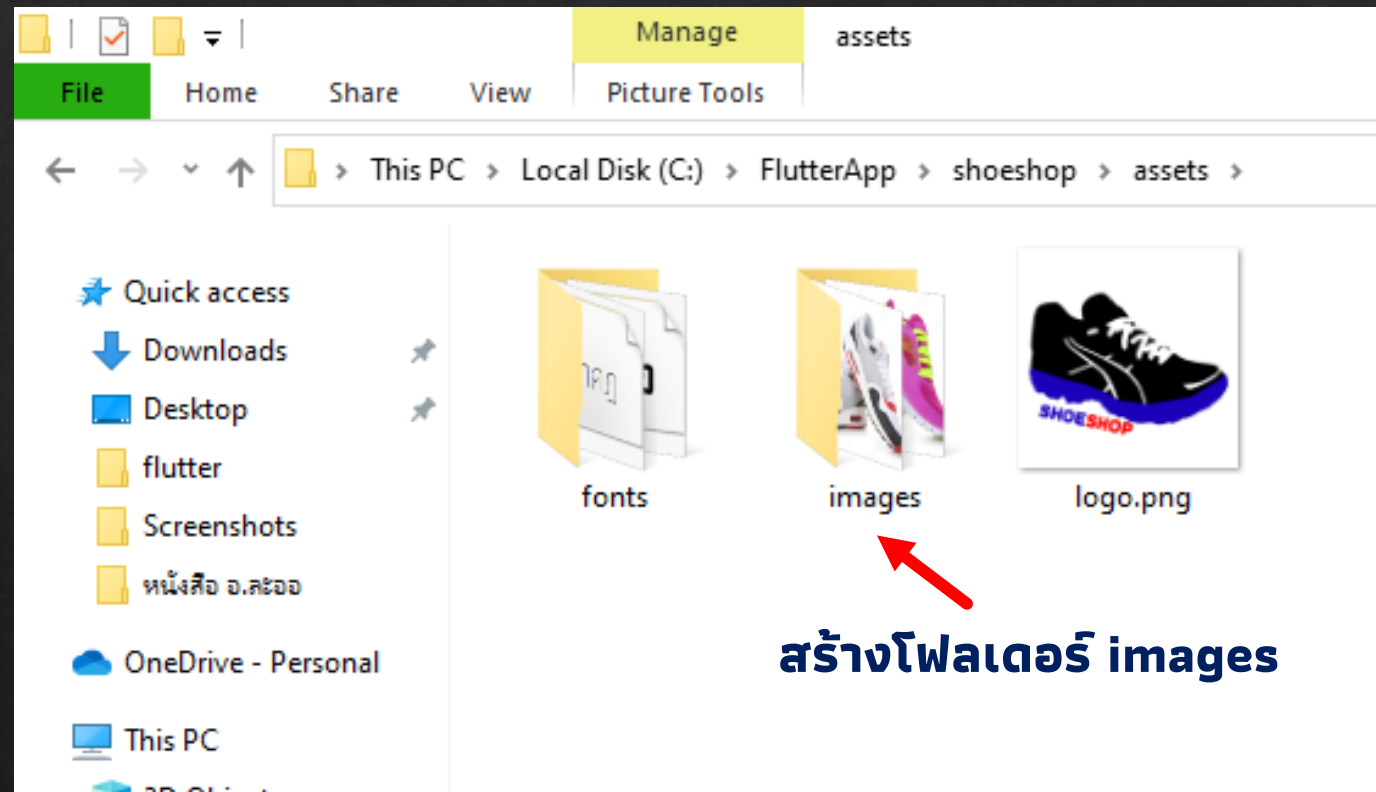
# Image

เป็นวิดเจ็ตที่ใช้แสดงรูปภาพ มีลักษณะพิเศษ ในการใช้งานที่แตกต่างกัน  
ชนิดของไฟล์รูปภาพที่รองรับการใช้งานมีหลายรูปแบบ

- image.asset
- image.network

# Image.asset

เป็นรูปแบบพื้นฐานที่ใช้แสดงผลรูปภาพ และถูกจัดเตรียมไว้เป็นทรัพยากรในระบบที่มีแอปพลิเคชันสามารถเรียกใช้งานได้



## ทำการแก้ไขไฟล์ pubspec.yaml

assets:

- assets/images/1.jpg
- assets/images/2.jpg
- assets/images/3.jpg
- assets/images/4.jpg
- assets/images/5.jpg



## เรียกใช้งาน

```
child: Image.asset(  
  "assets/images/2.jpg",  
  width: 150,  
),
```

# Image.network

เป็นรูปแบบแสดงข้อมูลที่ไ้ได้จากตำแหน่งจัดเก็บบนเครือข่ายอินเทอร์เน็ต

```
body: Column(  
  children: <Widget>[  
    Row(  
      children: <Widget>[  
        Image.network('ที่อยู่ภาพ')  
      ],  
    ),  
  ],  
);
```

# Center

เป็นวิดเจ็ต ที่ช่วยในการจัดแนวการแสดงผลของเนื้อหาใน Body ให้อยู่กึ่งกลาง

```
body: Center(  
  child: Container(  
    child: Image.asset('asset/images/c1.jpg')  
  ),  
)
```

# Padding

เป็นวิดเจ็ต ที่ทำหน้าที่สำหรับแทรกวิดเจ็ตลูก โดยการเติมช่องว่างหรือเพิ่มระยะห่างไว้รอบๆ ภายในพื้นที่ๆ กำหนดด้วยค่าของ EdgeInsets

- `EdgeInsets.all`
- `EdgeInsets.fromLTRB`

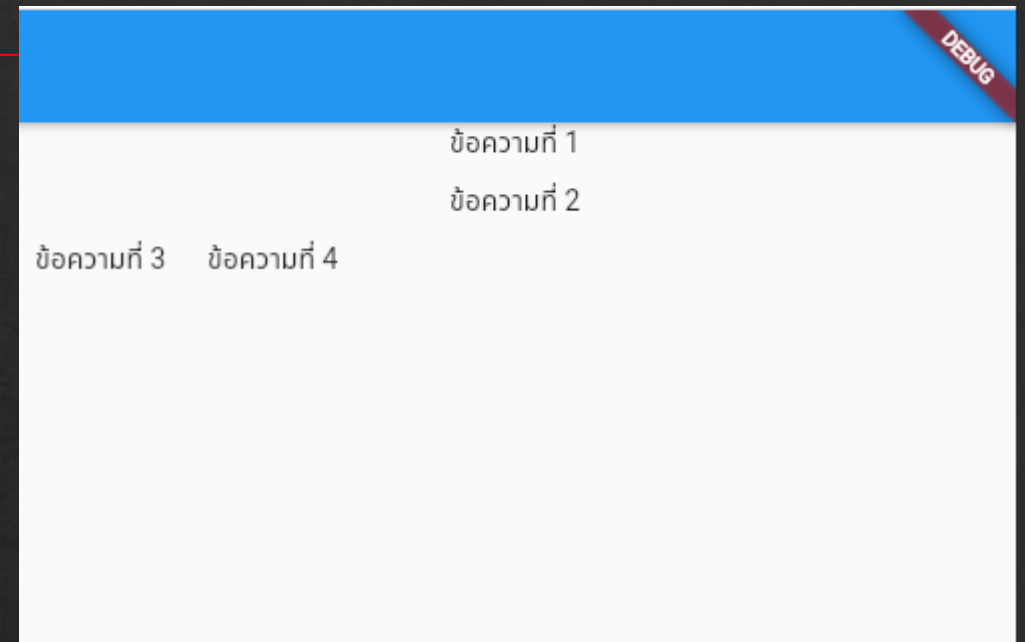
## **EdgeInsets.all**

เป็นการเว้นระยะห่างจากขอบของวัตถุรอบทุกด้านทั้ง 4 ด้าน คือ ขอบด้านบน ซ้าย ล่าง และขวา โดยกำหนดเป็นระยะด้วยค่าตัวเลขเป็นพิกเซลรองรับการกำหนดค่าเลขจำนวนเต็มและเลขจำนวนจริง เช่น `EdgeInsets.all(10)`

## **EdgeInsets.fromLTRB**

เป็นการเว้นระยะห่าง โดยกำหนดระยะห่างจากขอบแบบที่เป็นค่าอิสระแยกจากกัน ค่าที่กำหนดให้เป็นพารามิเตอร์ จะต้องกำหนดทั้งหมด 4 ค่า โดยเรียงลำดับจากขอบด้านซ้าย บน ขวา และล่าง ตามลำดับ เช่น `EdgeInsets.fromLTRB(10,5,7,0)`

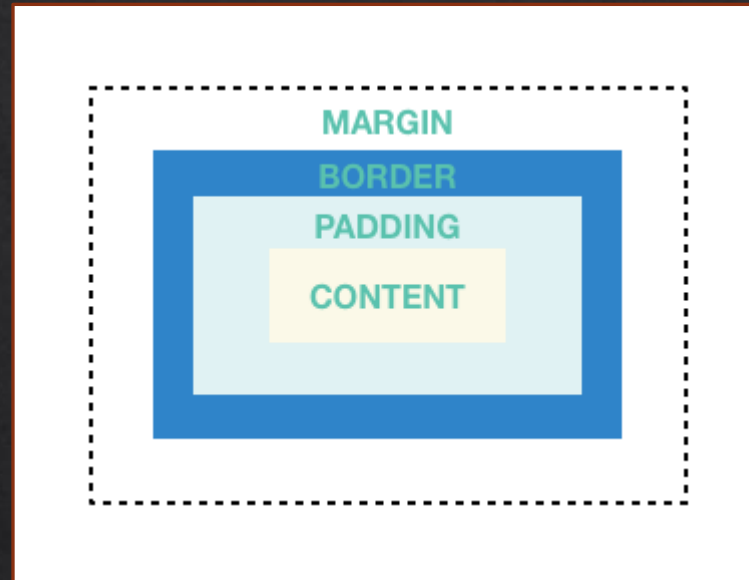
```
body: Column(  
  children: <Widget>[  
    Text('ข้อความที่ 1'),  
    Padding(  
      padding: const EdgeInsets.all(10),  
      child: Text('ข้อความที่ 2'),  
    ),  
    Row(  
      children: <Widget>[  
        Padding(  
          padding: const EdgeInsets.fromLTRB(10, 0, 20, 0),  
          child: Text('ข้อความที่ 3'),  
        ),  
        Text('ข้อความที่ 4'),  
      ],  
    ),  
  ],  
),
```



# Container

คือ Widget ที่ใช้กำหนดพื้นที่ สามารถกำหนด ขนาด รูปร่าง หรือจัดวาง ตำแหน่ง รวมไปถึงการกำหนดลวดลายและตกแต่งด้วยสีให้สวยงามได้ ปกติ ตัว Container จะไม่แสดงผลขึ้นมาให้เห็น เราจะต้องกำหนดความสูงลงไป และขนาดของ Container ก็จะไปขยายเต็มพื้นที่ ใช้ในการจัดวาง UI ให้แสดงผลเหมาะสมกับหน้าจอ

ตัว Container มี Widget ภายในสามารถแบ่งออกเป็น Layer ได้แบบนี้



เรียงลำดับที่อยู่ของ Properties จากด้านนอกเข้าสู่ด้านในจะเป็น  
Margin > Border > Padding > Content

ตัว Content นั้นจะถูกกำหนดโดย Properties ที่ชื่อ child ซึ่งสามารถบรรจุ Widget  
ได้หลายชนิด เช่น Row, Column, Image หรือว่าเป็น Container เอง แต่บรรจุได้แค่ตัวเดียว





```
21 | child: Container(  
22 |   color: Colors.red[600],  
23 |   child: Container(  
24 |     margin: const EdgeInsets.all(20.0),  
25 |     padding: const EdgeInsets.all(10.0),  
26 |     color: Colors.blue[600],  
27 |     child: Image.asset(  
28 |       'assets/images/logo.png'  
29 |     ) // Image.asset  
30 |   ) // Container  
31 | ) // Container
```

**margin** กำหนดให้มีช่องว่างเกิดขึ้นนอกกรอบของ widget container

**padding** เพิ่มระยะห่าง หรือช่องว่างทุกๆ ด้านของ widget ที่อยู่ใน widget container

**alignment** กำหนดตำแหน่งของการแสดงข้อความให้อยู่กึ่งกลาง alignment : Alignment.center

```
Center[
  child: Container[
    margin: const EdgeInsets.all(10.0),
    color: Colors.amber[600],
    width: 48.0,
    height: 48.0,
  ],
]
```

# ListView

เป็นเครื่องมือจัดรูปแบบการแสดงผลที่ทำให้วิดเจ็ตลูก (Children) สามารถเลื่อนไปมาตามทิศทางที่ต้องการได้ ช่วยให้สามารถจัดรูปแบบการแสดงผลวิดเจ็ตจำนวนมากๆ ให้อยู่ภายในขอบเขตตามต้องการและไม่เกิดข้อผิดพลาดในการแสดงผล


```
class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('ListView'),
      ), // AppBar
      drawer: Container(
        width: 250,
        color: Colors.red,
      ), // Container
      body: SafeArea(
        child: ListView(
          children: <Widget>[
            Container(
              height: 250,
              color: Colors.red,
            ), // Container
            Container(
              height: 250,
              color: Colors.green,
            ), // Container
            Container(
              height: 250,
              color: Colors.blue,
            ), // Container
            Container(
              height: 250,
              color: Colors.amber,
            ), // Container
          ], // <Widget>[]
        ), // ListView
      ), // SafeArea
    ); // Scaffold
  }
}
```





#1

☰ Menu 🔔

Pasta Pizza Salad Meat

 Chicago Pizza  
**\$9.99**  
Add

 Greek Pizza  
**\$9.99**  
Add

 Sicilian Pizza  
**\$9.99**  
Add



**EAT BIG!**

# สินค้า

DEBUG



F1  
2999 Bath



M1  
2999 Bath



F2  
2999 Bath





# Reference

- ◇ KongRuksiam Official. <https://www.youtube.com/c/KongRuksiamOfficial>
- ◇ เกรรินทร์ วกัญญเลิศสกุล. [2563]. พัฒนา Mobile App ด้วย Flutter & Dart. โปรวีชั่น, บจก.
- ◇ จีราวุธ วารินทร์. [2564]. ต่อยอดพัฒนาโมบายล์แอปด้วย Flutter + Firebase. ซิมพลีฟาย, สนพ.